IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Applicant:	Paul L. Sinclair, et al.	Group Art Unit:	2166
Serial No.:	09/713,887	Examiner:	Mohammad Ali
Filed:	November 16, 2000		
For:	LOCKING DATA IN A DATABASE AFTER AN OPERATION HAS BEGUN	Atty. Dkt. No.:	8779 (NCR)

Mail Stop: Appeal Brief
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

BRIEF IN SUPPORT OF APPEAL

Dear Sir:

This is a brief in support of Applicant's Notice of Appeal filed on June 8, 2007, in response to the final rejection dated March 8, 2007, in this matter. Filed contemporaneously with the Notice of Appeal was a pre-appeal brief and request for a Panel Review. On June 21, 2007, a Notice of Panel Decision was issued stating the case should proceed to the Board of Patent Appeals and Interferences. Therefore, Applicant is filing this brief along with any required fee(s).

(1) REAL PARTY IN INTEREST

The real party in interest in this matter is NCR Corporation, Dayton, Ohio, by virtue of an assignment recorded at reel 011339, frame 581-082, on November 16, 2000.

(2) RELATED APPEALS AND INTERFERENCES

Applicant is aware of no other active appeals or interferences related to this application.

(3) STATUS OF CLAIMS

Claims 1, 3-7, 9-15, 17-21 and 30 are currently pending in this application. Independent claims 1, 7, 15, 21, and 30, having been rejected two or more times, are under appeal. The text of the claims, as currently pending, is attached as an appendix to this brief.

(4) STATUS OF AMENDMENTS

On June 8, 2007, Applicant filed a notice of appeal with a pre-appeal brief in response to the final rejection dated March 8, 2007. In a Notice of Panel Decision from Pre-Appeal Brief Review mailed on June 21, 2007, the Panel rejected Applicant's arguments and stated the case should proceed to the Board of Patent Appeals and Interferences.

(5) SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 recites a method for use in managing data in a database system. The method comprising receiving a request to perform an operation on a set of target data residing in the database (e.g., page 8, lines 5-7; Fig. 2, element 200); executing the operation in the database on the set of target data (e.g., page 8, lines 9-13; Fig. 2, elements 210 and 215); during the execution of the operation, placing an initial lock on the target data to prevent concurrent execution of at least one operation on the target data (e.g., page 8, lines 14-18; Fig. 2, element 220); and during the executing of the operation, placing a final lock on the target data at a level that prevents concurrent

execution of a larger set of operations (e.g., page 8, line 19 – page 9, line 7; Fig. 2, elements 225-260).

Independent claim 7 recites a database system comprising at least one storage device (e.g., page 3, lines 6-8; Fig. 1, elements 110_1-110_M); at least one computing node configured to deliver data to and retrieve data from the storage device (e.g., page 3, lines 6-8; Fig. 1, elements 110_1-110_M and 105_1-105_N); and a database-management component configured to (e.g., page 3, lines 11-12; Fig. 1, elements 120_1-120_M): receive a request to perform an operation on a set of target data residing in the database (e.g., page 8, lines 5-7; Fig. 2, element 200); execute the operation in the database on the set of target data (e.g., page 8, lines 9-13; Fig. 2, elements 210 and 215); during the execution of the operation, place an initial lock on the target data to prevent concurrent execution of at least one but not all operations on the target data (e.g., page 8, lines 14-18; Fig. 2, element 220); and during the executing of the operation, place a final lock on the target data at a level that prevents concurrent execution of a larger set of operations (e.g., page 8, line 19 – page 9, line 7; Fig. 2, elements 225-260).

Independent claim 15 recites a computer program, stored on at least one computer-readable storage medium, for use in managing data in a database system, comprising executable instructions that, when executed by a computer, cause the computer to (e.g., page 10, lines 3-8; Fig. 1, element 100): receive a request to perform an operation on a set of target data residing in the database (e.g., page 8, lines 5-7; Fig. 2, element 200); execute the operation in the database on the set of target data (e.g., page 8, lines 9-13; Fig. 2, elements 210 and 215); during the executing of the operation, place an initial lock on the target data to prevent concurrent execution of at least one but not all operations on the target data (e.g., page 8, lines 14-18; Fig. 2, element 220); and during the executing of the operation, place a final lock on the target data at a level that prevents concurrent execution of a larger set of operations (e.g., page 8, line 19 – page 9, line 7; Fig. 2, elements 225-260).

Independent claim 21 recites a method for use in managing data in a database system, comprising: receiving an instruction from a user to perform a data-definition operation on a set of target data residing in the database (e.g., page 3, lines 20-22; page 5, lines 11-16; page 8, lines 5-7; Fig. 2, element 200); placing an initial lock on the target

data at a level that allows at least one concurrent operation on the target data (e.g., page 5, lines 11-14; page 8, lines 14-18; Fig. 2, element 220); executing the operation in the database on the set of target data (e.g., page 8, lines 9-13; Fig. 2, elements 210 and 215); during the execution of the operation, placing an initial lock on the target data to prevent concurrent execution of at least one but not all operations on the target data (e.g., page 8, lines 14-18; Fig. 2, element 220); and during the execution of the operation, placing a final lock on the target data at a level that excludes all other concurrent operations on the target data (e.g., page 8, line 19 – page 9, line 7; Fig. 2, elements 225-260).

Independent claim 30 recites a method for use in managing data in a database system, comprising: receiving an instruction from a user to perform a data-definition operation on a set of target data (e.g., page 3, lines 20-22; page 5, lines 11-16; page 8, lines 5-7; Fig. 2, element 200); placing an initial lock on the target data at a level that prevents at least one but not all types of concurrent operation on the target data (e.g., page 5, lines 1-16; Fig. 2, element 220); initiating execution of the operation on the target data; and during the execution of the operation, placing a final lock on the target data at a level that excludes all types of concurrent operations on the target data (e.g., page 5, lines 4-10).

(6) GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Independent claims 1, 7, 15, 21 and 30 are rejected under 35 USC § 102(b) as being anticipated by Lomet et at. (U.S. Patent No. 5,487,607; hereinafter "Lomet").

(7) ARGUMENT

The rejected independent claims and the claims that depend from them should be allowed over the cited reference for the reasons set forth below.

A 35 USC § 102(b) Rejection of Claims 1, 7, 15, 21 and 30 by Lomet

Lomet does not show or suggest "receiving a request to perform an operation on a set of target data residing in the database … during the executing of the operation,

placing an initial lock on the target data … and during the executing of the operation, placing a final lock on the target data at a level that prevents concurrent execution of a larger set of operations," as required by Applicant. Lomet teaches a method to improve concurrent execution of transactions by using range locking and key locking of data in a database. (Abstract; Col. 4, lines 7-14; and Col. 4, lines 33-36) Lomet teaches that a database receives a first request to place a lock on a range of key values. The request is processed by the database and it completes if the database determines that the lock can be placed on the requested range of key values. At some later time, a second request is received by the database to place a lock on a single key value that is in the same range of key values as the first lock. The second request is processed by the database and it completes if the database determines that the lock can be placed on the single key. (Col. 6, line 53 – col. 7, line 14). Lomet teaches that the first lock is placed on a range of data and that the second lock is placed on a single key value. This is different from what is required by Applicant. Applicant requires placing both the initial and final locks on the same target data. Lomet teaches that the first lock is placed on a range of key values and that the second lock is placed on a single key value. Clearly, Lomet's locks are placed on two different sets of data and not the same target data required by Applicant. Thus, at least this required element of Applicant's claimed invention is missing from Lomet.

Lomet further teaches that transactions perform multiple operations and that each operation can make a request to the database for a different lock. Lomet states "each transaction routine comprising instructions that implement database-access operations, associated with respective database accesses, that apply lock requests to the lock manager…." (Col. 17, lines 34-36.) Thus, Lomet teaches the use of multiple database operations and that each operation can make a request to apply a lock to data in a database. Applicant on the other hand, requires receiving a single request to perform an operation on a set of target data and further requires placing both the initial lock and the final lock on the target data during the execution of the single operation. Lomet does not teach that a single operation causes the database to issue multiple locks. Lomet teaches the use of multiple operations that are sent to the database separately for execution by the database. In the above example by Lomet, a first operation is sent to the database and requests a first lock on a range of key values. Then a second operation is sent to the

database and requests a second lock a single key value. These are all separate operations in Lomet's teaching. Applicant requires a single operation that places at least two locks on the target data. Thus, at least this required element of Applicant's claimed invention is missing from the teaching of Lomet.

Anticipation under 35 U.S.C. §102 is established only when a single prior art reference discloses each and every element of the claimed invention. RCA Corp. v. Applied Digital Data Systems, Inc., 221 USPQ 385 (Fed Cir. 1984). The rejection of Applicant's claim as being anticipated by Lomet is thus improper and Applicant asks the Board to reverse this rejection.

B. Conclusion

Lomet does not show or suggest all of the elements of Applicant's claimed invention. Therefore, the rejection is improper. Applicant asks the Board to reverse the rejection and to allow all claims.

Please apply any charges or credits that might be due, except the issue fee, to the NCR Corporation deposit account number 14-0225.

Respectfully submitted,

Date September 21, 2007

(Electronically Filed)

/Harden E. Stevens, III/

Harden E. Stevens, III
Agent for Applicant
Reg. No. 55,649

NCR Corporation
Law Department
1700 South Patterson Blvd.
Dayton, Ohio 45479

Tel. No. (803) 939-6505
Fax No. (803) 939-5099
Email: ss133111@ncr.com

(8) APPENDIX - Listing of Current Claims

Claim 1: (Previously amended) A method for use in managing data in a database system, comprising:

receiving a request to perform an operation on a set of target data residing in the database;

executing the operation in the database on the set of target data;

during the execution of the operation, placing an initial lock on the target data to prevent concurrent execution of at least one operation on the target data; and

during the executing of the operation, placing a final lock on the target data at a level that prevents concurrent execution of a larger set of operations.

Claim 2: (Canceled)

Claim 3: (Previously amended) The method of claim 1, where the initial lock allows concurrent execution of operations that involve reading the target data.

Claim 4: (Previously amended) The method of claim 1, where the final lock prevents concurrent execution of all operations on the target data.

Claim 5: (Previously amended) The method of claim 1, further comprising allowing a user to specify the type of lock initially placed on the data.

Claim 6: (Original) The method of claim 1, where the operation is one of the following types: a COLLECT STATISTICS operation, a CREATE INDEX operation, and an ALTER TABLE operation.

Claim 7: (Previously amended) A database system comprising:

at least one storage device;

at least one computing node configured to deliver data to and retrieve data from the storage device; and

a database-management component configured to:

receive a request to perform an operation on a set of target data residing in the database;

execute the operation in the database on the set of target data;

during the execution of the operation, place an initial lock on the target data to prevent concurrent execution of at least one but not all operations on the target data; and

during the executing of the operation, place a final lock on the target data at a level that prevents concurrent execution of a larger set of operations.

Claim 8: (Cancel)

Claim 9: (Previously amended) The system of claim 7, where the initial lock allows concurrent execution of at least one other operation on the target data.

Claim 10: (Previously amended) The system of claim 7, where the subsequent lock prevents concurrent execution of all other operations on the target data.

Claim 11: (Previously amended) The system of claim 7, where the database-management system is configured to allow a user to specify the type of lock initially placed on the data.

Claim 12: (Original) The system of claim 7, comprising multiple computing nodes and multiple storage devices, where each storage node is configured to manage storage of data on at least a subset of the storage devices.

Claim 13: (Previously amended) The system of claim 12, where the database-management system is configured to place locks on a block of data that is spread across more than one of the storage devices.

Claim 14: (Original) The system of claim 7, where the operation is one of the following types: a COLLECT STATISTICS operation, a CREATE INDEX operation, and an ALTER TABLE operation.

Claim 15: (Previously amended) A computer program, stored on at least one computer-readable storage medium, for use in managing data in a database system, comprising executable instructions that, when executed by a computer, cause the computer to:

receive a request to perform an operation on a set of target data residing in the database;

execute the operation in the database on the set of target data;

during the executing of the operation, place an initial lock on the target data to prevent concurrent execution of at least one but not all operations on the target data; and

during the executing of the operation, place a final lock on the target data at a level that prevents concurrent execution of a larger set of operations.

Claim 16: (Cancel)

Claim 17: (Previously amended) The program of claim 15, where the initial lock allows concurrent execution of at least one other operation on the target data.

Claim 18: (Previously amended) The program of claim 15, where the subsequent lock prevents concurrent execution of all other operations on the target data.

Claim 19: (Previously amended) The program of claim 15, where the program causes the computer to allow a user to specify the type of lock initially placed on the data.

Claim 20: (Original) The program of claim 15, where the operation is one of the following types: a COLLECT STATISTICS operation, a CREATE INDEX operation, and an ALTER TABLE operation.

Claim 21: (Previously amended) A method for use in managing data in a database system, comprising:

receiving an instruction from a user to perform a data-definition operation on a set of target data residing in the database;

placing an initial lock on the target data at a level that allows at least one concurrent operation on the target data;

executing the operation in the database on the set of target data;

during the execution of the operation, placing an initial lock on the target data to prevent concurrent execution of at least one but not all operations on the target data; and

during the execution of the operation, placing a final lock on the target data at a level that excludes all other concurrent operations on the target data.

Claim 22: (Cancel)

Claim 23: (Original) The method of claim 21, further comprising allowing a user to select the level of the initial lock.

Claim 24: (Original) The method of claim 21, where placing an initial lock on the target data includes placing one of the following types of locks on the target data: an ACCESS lock; a READ lock; and a WRITE lock.

Claim 25: (Original) The method of claim 21, where placing a final lock on the target data includes placing an EXCLUSIVE lock on the target data.

Claim 26: (Original) The method of claim 21, where placing an initial lock on the target data includes locking an entire table.

Claim 27: (Original) The method of claim 21, where receiving the instruction from the user includes receiving an instruction to perform one of the following operations: a CREATE INDEX operation, a COLLECT STASTICS operation, and an ALTER TABLE operation.

Claim 28: (Canceled)

Claim 29: (Canceled)

Claim 30: (Previously amended) A method for use in managing data in a database system, comprising:

receiving an instruction from a user to perform a data-definition operation on a set of target data;

placing an initial lock on the target data at a level that prevents at least one but not all types of concurrent operation on the target data;

initiating execution of the operation on the target data; and

during the execution of the operation, placing a final lock on the target data at a level that excludes all types of concurrent operations on the target data.

(9) EVIDENCE APPENDIX

None

(10) RELATED PROCEEDINGS APPENDIX

None